



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2003-0003335
Application Number

출원 년 월 일 : 2003년 01월 17일
Date of Application
JAN 17, 2003

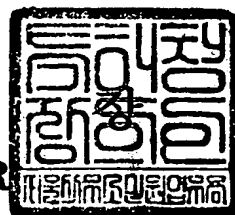
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 06 월 03 일

특 허 청

COMMISSIONER



【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2003.01.17
【국제특허분류】	H04L
【발명의 명칭】	직접 메모리 접근매체의 데이터 전송 장치 및 방법
【발명의 영문명칭】	APPARATUS AND METHOD FOR DATA TRANSMISSION IN DMA
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	2003-001449-1
【발명자】	
【성명의 국문표기】	정문기
【성명의 영문표기】	CHUNG, Moon Kee
【주민등록번호】	780202-1901219
【우편번호】	630-492
【주소】	경상남도 마산시 회원구 양덕2동 73-34
【국적】	KR
【발명자】	
【성명의 국문표기】	박정호
【성명의 영문표기】	PARK, Jung Ho
【주민등록번호】	750806-1496215
【우편번호】	442-470
【주소】	경기도 수원시 팔달구 영통동 990-3 가람주택 304호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이건주 (인)

【수수료】

【기본출원료】	20	면	29,000	원
【가산출원료】	16	면	16,000	원
【우선권주장료】	0	건	0	원
【심사청구료】	14	항	557,000	원
【합계】	602,000	원		

【요약서】**【요약】**

본 발명은 메모리에 저장되어 있는 데이터를 다른 메모리로 이동시키는 장치 및 방법에 관한 것으로서, 특히 데이터 이동시 특정 비트 수만큼 쉬프트를 수행한 후 이동시키는 장치 및 방법에 관한 것이다. 기존의 직접 메모리 접근매체(DMA)은 제 1 메모리에서 데이터를 읽어서 제 2 메모리 메모리 어드레스의 첫 번째 위치부터 차례로 저장한다. 하지만 본 발명에서는 DMA 내의 제어 레지스터에 데이터 이동시 특정 비트 수만큼 쉬프트를 수행한 후 데이터를 이동시키도록 레지스터 값들을 설정한다. 이와 같이 함으로서 제어 레지스터에 설정된 값만큼 쉬프트된 데이터를 상기 다른 메모리에 저장된다.

【대표도】

도 6

【색인어】

DMA, 버스 중재기, 비트 쉬프트, AMBA

【명세서】

【발명의 명칭】

직접 메모리 접근매체의 데이터 전송 장치 및 방법{APPARATUS AND METHOD FOR DATA TRANSMISSION IN DMA}

【도면의 간단한 설명】

도 1은 일반적인 암바(AMBA) 구조를 도시한 도면.

도 2는 일반적인 직접 메모리 접근매체(DMA)의 구조를 도시한 도면.

도 3은 본 발명이 적용되는 엔디안 변환 과정에 나타난 도면.

도 4는 본 발명이 적용되는 데이터 전송시 비트 쉬프트가 수행되는 과정을 도시한 도면.

도 5는 본 발명이 적용되는 DMA의 제어 레지스터의 레지스터 값을 설정하는 과정을 나타낸 흐름도.

도 6은 본 발명에 의한 제1 저장장치에서 제 2저장장치로 데이터 전송 과정을 나타낸 도면.

도 7은 본 발명을 이용하여 프로토콜 계층에서 데이터를 쉬프트하는 과정을 나타낸 도면.

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <8> 본 발명은 데이터 저장장치 및 방법에 관한 것으로서 특히, 특정 메모리에 저장되어 있는 데이터를 다른 저장 장소로 이동시키는 장치 및 방법에 관한 것이다.
- <9> 일반적으로 데이터의 이동에는 중앙 처리 장치(central processing unit: 이하 CPU라 한다.)의 명령에 의하여 이루어진다. 상기 CPU는 시스템 전체를 제어하는 장치로서, 다양한 입력장치들로부터 자료를 받아서 처리한 후 그 결과를 출력장치로 보내는 일련의 과정을 제어하고 조정하는 일을 수행한다. 이와 같이 상기 CPU가 복수개의 입력 장치들로부터 데이터를 입력받아 처리하는 경우, 상기 CPU 내부에는 많은 부하가 발생하게 된다. 따라서 상기 CPU 기능의 일부를 다른 프로세스에 의해 수행됨이 요구된다. 상기 요구에 의해 직접 메모리 접근매체(Direct Memory Access : 이하 DMA라 한다.) 프로세스가 개발되었다.
- <10> 상기 DMA는 상기 CPU의 기능 중 특정 메모리나 저장 장소에 저장되어 있는 데이터들을 독출하여 다른 메모리나 저장장소에 전송하는 기능을 수행한다. 즉, 크기가 작은 데이터는 상기 CPU에 의해 상기 특정 메모리에서 상기 다른 메모리로 전송하여도 상기 CPU에 많은 부하가 발생하지 않는다. 그러나 일정 크기 이상을 가지는 데이터의 경우에는 상기 CPU에 의해 상기 특정 메모리에서 상기 다른 메모리로 전송하면, 상기 CPU에 많은 부하가 발생한다. 또한 상기 일정 크기 이상을 가지는 데이터의 경우 상기 CPU에 의

한 데이터 전송속도보다 상기 DMA의 의한 데이터 전송속도가 높아진다. 일반적으로 상기 데이터의 크기는 512byte 정도가 된다.

<11> 도 1은 일반적인 DMA를 사용하는 시스템의 예로서, AMBA(Advanced Micro-controller Bus Architecture)의 구조를 도시한 도면이다. 상기 AMBA는 ARM(Advanced Risc Machine)사에서 제조되었다. 상기 AMBA는 높은 주파수가 요구되는 블록에서 사용되는 AHB(Advanced High-performance Bus)와 상대적으로 낮은 주파수에서 사용되는 APB(Advanced Peripheral Bus)가 포함된다. 상기 AHB와 상기 APB는 AHB-APB 브릿지(Bridge)(108)로 연결되어 있어 고속의 버스와 저속의 버스가 상호 데이터를 주고 받을 수 있다. 상기 도 1의 나머지 구성을 살펴보면 상기 AHB 블록에는 중앙처리장치(CPU)(100)와 제 1저장장치(102), DMA 장치(104), 그리고 버스 중재기(Bus Arbiter)(106)로 구성되어 있다. 또한 상기 APB 블록에는 제 2저장장치(110)와 입출력 장치(112)로 장치로 구성되어 있다. 상기 입출력 장치에는 유니버설 시리얼 버스(Universal Serial Bus :이하 USB라 한다.)나 키 패드(Keypad), 범용 비동기화 송수신기(Universal Asynchronous Receiver/Transmitter : 이하 UART라 한다.)등이 포함될 수 있다. 종래에는 상기 제 1저장장치(102)에 저장되어 있는 데이터를 상기 제 2 저장장치(110)로 전송하기 위해서는 상기 CPU(104)에 의해 모든 동작이 제어되고 처리된다. 즉, 상기 CPU(104)는 상기 제 1저장장치(102)에 저장되어 있는 데이터를 상기 제 2 저장장치(110)로 전송할 필요가 있는 경우, 상기 제 1 저장장치(102)에 저장되어 있는 데이터를 독출한다. 상기 데이터를 독출한 상기 CPU(100)는 상기 제 2저장장치(110)로 전송하기 이전에 필요한 처리과정을 거친다. 하지만 상기 제 1저장장치(102)에서 상기 제 2저장장치(110)로 전송할 데이터 양이 많은 경우 상기 CPU(100)가 이와 같은 일련의 과정을 수

행하기가 곤란한 문제가 발생한다. 즉, 상기 CPU(100)는 데이터의 전송 이외에 시스템에 관련된 많은 작업들을 수행하여야 하기 때문이다. 따라서 이와 같은 문제점을 해결하기 위해 상기 DMA 장치가 개발되었다.

<12> 상기 도 1에서는 하나의 DMA 장치(104)만이 표시되어 있으나 상기 DMA 장치(104)는 복수 개로 구성될 수 있음은 자명하다. 또한 상기 DMA 장치(104)는 상기 AHB 블록에 저장되어 있는 데이터를 상기 AHB 블록으로 전송할 수도 있으며, 동시에 상기 APB 블록에 저장되어 있는 데이터를 상기 APB 블록으로 전송할 수도 있다. 이 경우 상기 AHB 블록과 상기 APB 블록은 복수 개의 저장장치로 구성된다. 상기 DMA(104)은 CPU(100)의 제어에 의해, 또는 외부의 제어에 의해 상기 제 1저장장치(102)에 저장되어 있는 데이터를 상기 제 2 저장장치(110)로 전송한다. 상기 DMA 장치(104)에서의 데이터 전송에 관해서는 이하 도 3에서 상세히 설명한다.

<13> 도 2는 상기 도 1의 DMA 장치(104)의 내부 구조를 도시한 블록도이다. 상기 도 2에서 보이고 있는 바와 같이 상기 DMA 장치(104)는 제어 레지스터(Control Register)와 소스 어드레스 레지스터(Source Address Register :이하 SAR 이라 한다.)와 목적지 어드레스 레지스터(Destination Address Register :이하 DAR 이라 한다.), 그리고 전송 카운터 레지스터(Transfer Count Register :이하 TCR 이라 한다.)와 버퍼(Frist in Frist out :FIFO 라 한다.)와 버스 제어기와 인터페이스 등으로 구성되어 있다. 상기 SAR은 상기 DMA(104)이 상기 제 1 저장장치(102)로부터 데이터를 독출할 경우에 있어 초기 소스 어드레스를 지정하는 레지스터이다. 상기 DAR은 상기 DMA(104)이 상기 제 1저장장치(102)로부터 독출한 데이터를 가장 먼저

상기 제 2 저장장치(110)에 기록하기 위한 초기 목적지 어드레스를 지정하는 레지스터이다. 상기 TCR은 상기 DMA(104)가 상기 제 1 저장장치(102)에서 독출한 데이터를 상기 제 2 저장장치(110)에 기록하는 회수를 지정하는 레지스터이다. 또한 상기 제어 레지스터는 상기 제 1 저장장치(102)의 소스 어드레스에서 데이터를 독출한 후 다음 데이터를 독출할 경우, 상기 독출할 어드레스를 감소시킬 것인지 증가시킬 것인지 또는 고정시킬 것인지 제어한다. 또한 상기 제어 레지스터는 상기 제 1 저장장치(102)의 소스 어드레스로부터 독출한 데이터를 목적지 어드레스에 기록한 후, 다음 데이터의 기록이 요구될 경우 상기 기록을 위한 어드레스를 감소시킬 것인지 증가시킬 것인지 또는 고정시킬 것인지 제어한다. 또한, 상기 제어 레지스터는 상기 제 1 저장장치(102)에서 상기 제 2 저장장치(110)로 데이터를 전송할 경우 한번에 전송 가능한 데이터의 단위를 제어한다. 상기 한번에 전송 가능한 데이터의 단위에는 1 바이트(byte)(8비트), 1/2-워드(word)(16비트), 1 워드(32비트)등이 있다. 상기 레지스터들에 등록되는 값들은 상기 CPU(100)의 명령에 의해 또는 외부 제어기의 명령에 의해 이루어짐은 자명하다.

<14> 상술한 바와 같이 상기 DMA 내부에 위치하고 있는 레지스터에 대한 값들이 설정되면 상기 도 1에 도시된 상기 제 1 저장장치(102)로부터 데이터를 독출하여 상기 제 2 저장장치(110)에 독출한 데이터를 기록한다.

<15> 이와 같이 상기 DMA는 상기 제 1 저장장치에 저장되어 있는 데이터를 상기 제 2 저장장치로 전달하는 기능만을 수행한다. 하지만 상기 제 2 저장장치에는 상기 제 1 저장장치에 저장되어 있는 데이터와 다른 형태로서 데이터가 저장될 것을

요구하는 경우가 발생한다. 예컨대, 상기 제 1 저장장치에 저장되어 있는 데이터를 일정 비트 수만큼 쉬프트(Shift)한 데이터가 상기 제 2 저장장치에 저장될 것을 원하는 경우가 발생한다. 이런 경우 종래에는 DMA에서 데이터를 비트 쉬프트 없이 이동시킨 후, CPU가 상기 이동된 데이터를 일정 비트 수만큼 쉬프트시켰다. 또는 상기 DMA를 사용하지 않고 상기 CPU가 데이터를 일정 비트 수만큼 쉬프트시킨 후 이동시켰다. 이와 같이 상기 데이터를 특정 비트 쉬프트 이동시킬 경우 상기 CPU에 의해 수행됨으로서 상기 DMA로 데이터를 이동시킬 경우에 비해 전송속도가 낮아지는 단점이 있다. 또한 상기 CPU의 일부 기능을 상기 DMA에서 처리함으로서 상기 CPU의 부하를 낮추고자 하였으나 이와 같이 데이터를 특정 비트 쉬프트시키는 경우 상기 CPU가 관여함으로서 부하가 낮추어지지 않게 된다. 따라서 상기 CPU의 관여없이 상기 데이터를 특정 비트 쉬프트 이동시킬 필요가 요구된다.

【발명이 이루고자 하는 기술적 과제】

- <16> 따라서, 전술한 종래 기술의 문제점을 해결하기 위한 본 발명의 목적은 제 1 저장장치로부터 제 2 저장장치로부터 데이터 전송 시 중앙처리장치의 관여없이 특정 비트수만큼 쉬프트된 데이터를 전송하는 장치 및 방법을 제안함에 있다.
- <17> 본 발명의 다른 목적은 종래 중앙처리장치에 의해 데이터를 특정 비트수만큼 쉬프트시켜 전송함으로서 발생하는 처리 시간의 지연을 방지하는 장치 및 방법을 제안함에 있다.

<18> 상기한 본 발명의 목적을 이루기 위해 직접 메모리 접근매체(DMA)의 제어 레지스터에 쉬프트를 원하는 특정 비트 수와 쉬프트 방향을 결정하는 레지스터 값을 설정한다. 상기 레지스터 값이 설정이 완료되면, 제 1 저장장치로부터 데이터를 독출하여 상기 DMA의 임시 저장장치에 저장한다. 상기 DMA의 임시 저장장치에 저장된 데이터를 독출하여 상기 설정된 쉬프트를 원하는 특정 비트 수와 쉬프트 방향에 따라 쉬프트를 수행한 후 제 2 저장장치에 저장하는 장치를 제안한다.

<19> 상기한 본 발명의 목적을 이루기 위해 직접 메모리 접근매체(DMA)의 제어 레지스터에 쉬프트를 원하는 특정 비트 수와 쉬프트 방향을 결정하는 레지스터 값을 설정한다. 상기 레지스터 값이 설정이 완료되면, 제 1 저장장치로부터 데이터를 독출하여 상기 DMA의 임시 저장장치에 저장한다. 상기 DMA의 임시 저장장치에 저장된 데이터를 독출하여 상기 설정된 쉬프트를 원하는 특정 비트 수와 쉬프트 방향에 따라 쉬프트를 수행한 후 제 2 저장장치에 저장하는 방법을 제안한다.

【발명의 구성 및 작용】

<20> 이하 본 발명이 바람직한 실시 예를 첨부한 도면의 참조와 함께 상세히 설명한다. 또한 본 발명을 설명함에 있어서, 관련된 공지기능 혹은 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단된 경우 그 상세한 설명은 생략한다.

<21> 상술한 바와 같이 상기 도 2는 상기 DMA의 블록 구성도이다. 또한 상기 CPU(100)는 외부 제어장치 의해 상기 SAR, DAR, TCR 등의 값들이 지정된다. 하지만 본 발명에서는 상기 SAR, DAR, TCR 등의 값 이외에 상기 데이터를 전달함에 있어 특정 비트만큼

쉬프트 시키기 위한 특정 값을 지정한다. 먼저 본 발명에서는 DMA에서 이동시키고자 하는 데이터를 처리할 수 있도록 추가적인 정보들이 요구된다. 기존에 데이터를 이동시키기 위해 요구되는 정보와 함께 상기 데이터를 처리하기 위한 정보들의 예는 하기 표 1에서 보이고 있는 바와 같다. 이하 하기 표 1은 본 발명이 적용되는 제어 레지스터의 구성과 상기 각 구성의 역할에 대해 상세하게 설명하고 있다.

<22> -【표 1】

제어 레지스터			
비트 수	명칭	역할	기능
11 내지 13	쉬프트 카운터	비트 쉬프트 할 비트 수 결정	쉬프트 카운터:0내지 7
10	쉬프트 방향	비트 쉬프트 할 방향 결정	0 : 오른쪽으로 쉬프트 1 : 왼쪽으로 쉬프트
9	쉬프트 인에이블	비트 쉬프트 수행 여부 결정	0 : 쉬프트 수행 1 : 쉬프트 수행하지 않음
8	엔디안	엔디안 변환 여부 결정	0 : 엔디안 변환 수행하지 않음 1 : 엔디안 변환 수행
7	목적지 어드레스 방향	목적지 어드레스 증가/감소 여부 결정	0 : 증가 1 : 감소
6	소스 어드레스 방향	소스 어드레스 증가/감소 여부 결정	0 : 증가 1 : 감소
5	목적지 어드레스 고정	목적지 어드레스 고정 여부 결정	0 : 증가 또는 감소 1 : 고정
4	소스 어드레스 고정	소스 어드레스 고정 여부 결정	0 : 증가 또는 감소 1 : 고정
2 내지 3	전송 사이즈	데이터의 전송 비트 단위	00 : 8비트 01 : 16비트 10 : 32비트
1	DMA 모드	DMA의 모드 결정	0 : S/W 1 : H/W
0	DMA 구동	DMA의 구동 여부 결정	0 : DMA 수행되지 않음 1 : DMA 수행

<23> 상기 표 1에서 보이고 있는 바와 같이 상기 제어 레지스터는 모두 14비트들로 구성되어 있다. 또한 상기 각 비트들은 각각의 기능을 수행한다. 이하 상기 표

1에서 제안되고있는 본원 발명의 제어 레지스터의 기능에 대해 상세히 알아본다. 상기 각 비트의 값은 사용자의 선택에 의해 임의로 조정될 수 있음은 자명하다. 상기 제어 레지스터의 0번 비트는 상기 DMA(104)에 의한 데이터 전달 여부를 결정한다. 상기 0번 비트의 값이 "0"인 경우 상기 DMA(104)에 의한 데이터 전송을 수행하지 않으며, 상기 0번 비트의 값이 "1"인 경우 상기 DMA(104)에 의한 데이터 전송을 수행한다. 상기 DMA(104)에 의한 데이터 전송이 수행되지 않는 경우는 다음과 같다. 즉, 상기 전송할 데이터 양이 작아서 상기 CPU(100)에 의한 데이터 전송이 이루어지는 경우와 데이터 전송이 수행되지 않는 경우이다. 상기 0번 비트가 "1"인 경우 상기 레지스터의 1번 비트를 체크한다.

<24> 상기 제어 레지스터의 1번 비트는 상기 DMA(104) 모드를 결정하는 비트이다. 상기 DMA모드(Mode)란 상기 DMA(104)의 수행이 하드웨어(Hardware)에 의해 수행되는 경우인지, 소프트웨어(Software)에 의해 수행되는 경우인지 여부를 판단한다. 상기 소프트웨어에 의한 수행이란 상기 CPU(100)의 제어 명령에 의해 상기 DMA(104)에서 데이터 전송이 이루어지는 경우를 말하며, 상기 하드웨어에 의한 수행이란 외부 제어 시스템의 제어 명령에 의해 상기 DMA(104)에서 데이터 전송이 이루어지는 경우를 말한다. 상기 1번 비트의 값이 "0"인 상기 S/W에 의해 수행되는 경우이며, 상기 1번 비트의 값이 "1"인 경우 상기 H/W에 의해 수행되는 경우이다. 상기 제어 레지스터의 2번 비트 내지 3번 비트는 전송 사이즈 결정 비트이다. 즉, 한번에 수행되는 데이터 전송에 있어 전송 비트 단위를 결정한다. 상기 비트의 값이 "00"인 경우 8비트(1바이트) 단위로 데이터 전송이 수행되어지며, 상기 비트의

값이 "01"인 경우 16비트(1/2 워드) 단위로 데이터 전송이 수행되어진다. 또한 상기 비트의 값이 "10"인 경우 32비트(1 워드) 단위로 데이터 전송이 수행되어진다.

<25> 상기 제어 레지스터의 4번 비트는 소스 어드레스 고정(Source Address Fix : 이하 SAF라 한다.)여부를 결정하는 비트이다. 즉, 상기 도 2의 SAR이 지정하는 어드레스에 저장되어 있는 데이터를 독출한 후 다음 데이터를 독출할 어드레스를 결정하는 비트이다. 다음 데이터의 독출 어드레스를 증가/감소시킬 것인지, 동일한 어드레스에 저장되어있는 데이터를 반복해서 독출할 것인지 여부를 판단한다. 상기 4번 비트의 값이 "0"인 경우 상기 다음 데이터의 독출 어드레스를 증가/감소시키며, 상기 4번 비트의 값이 "1"인 경우 상기 다음 데이터는 이전에 독출한 어드레스와 동일한 어드레스에서 데이터를 독출한다. 상기 제어 레지스터의 5번 비트는 목적지 어드레스 고정(Destination Address Fix : 이하 DAF라 한다.)여부를 결정하는 비트이다. 즉, 상기 도 2의 DAR이 지정하는 목적지 어드레스에 상기 소스 어드레스로부터 독출한 데이터를 저장한 이후, 다음에 독출한 데이터를 저장할 목적지 어드레스를 결정하는 비트이다. 상기 다음에 독출한 데이터를 저장할 목적지 어드레스를 증가/감소시킬 것인지 또는, 동일한 어드레스에 반복해서 독출한 데이터를 기록할 것인지 여부를 판단한다. 상기 4번 비트의 값이 "0"인 경우 상기 독출한 다음 데이터의 목적지 어드레스를 증가/감소시킨 후 저장하며, 상기 4번 비트의 값이 "1"인 경우 상기 독출한 다음 데이터의 목적지 어드레스를 이전에 저장한 목적지 어드레스와 동일한 어드레스에 데이터를 기록한다.

<26> 상기 제어 레지스터의 6번 비트는 상기 4번 비트의 값이 "0"인 경우 상기 소

스 어드레스를 증가시킬 것인지 감소시킬 것인지 여부를 결정하는 (소스 어드레스 방향: Source Address Direction)비트 이다. 상기 이후 독출할 데이터의 소스 어드레스 값을 증가시킬 경우 그 값은 "0"이며, 상기 이후 독출할 데이터의 소스 어드레스 값을 감소시킬 경우 그 값은 "1"이다. 상기 제어 레지스터의 7번 비트는 상기 5번 비트의 값이 "0"인 경우 목적지 어드레스를 증가시킬 것인지 감소시킬 것인지 여부를 결정하는 (목적지 어드레스 방향: Destination Address Direction)비트 이다. 상기 이후 독출한 데이터를 기록할 목적지 어드레스 값을 증가시킬 경우 그 값은 "0"이며, 상기 이후 독출한 데이터를 기록할 목적지 어드레스 값을 감소시킬 경우 그 값은 "1"이다.

<27> 상기 제어 레지스터의 8번 비트는 엔디안(Endian)변환 결정 비트이다. 상기 엔디안 변환이란 상기 제어 레지스터의 2번 비트 내지 3번 비트에서 결정된 데이터의 전송 단위 비트가 결정된 후, 상기 전송 단위 비트가 32비트(1워드)인 경우에만 이용된다. 도 3은 상기 엔디안 변환 과정을 도시하고 있다. 상기 도 3에서 보이고 있는 바와 같이 상기 전송 비트 단위는 32비트(1워드)로 구성되어 있으며, 상기 1워드는 4개의 바이트로 구성되어 있다. 상기 4개의 바이트는 A, B, C, D이며, 이 중 가장 중요도가 높은 바이트는 A바이트이고 그 다음으로 중요한 바이트는 B 바이트이다. 또한 가장 중요도가 낮은 바이트는 D바이트이다. 하지만 상기 엔디안 변환을 수행하면 상기 바이트들의 중요도가 변환된다. 즉 가장 중요도가 높은 바이트가 가장 중요도가 낮은 바이트로 변환되며, 가장 중요도가 낮은 바이트가 가장 중요도가 높은 바이트로 변환된다. 상기 제어 레지스터의 8번 비트의 값이 "0"인 경우 상기 엔디안 변환을 수행하지 않으며, 상기 제어 레지스터의 8번 비트의 값이 "1"인 경우 상기 엔디안 변환을 수행한다.

<28> 상기 제어 레지스터의 9번 비트는 본 발명의 주요 내용인 비트 쉬프트를 결정하는 비트이다. 상기 비트 쉬프트는 상기 DMA의 내부에 위치하고 있는 상기 FIFO에서 수행되어진다. 상기 제 1저장장치의 소스 어드레스로부터 독출한 데이터는 상기 FIFO에 임시로 저장된다. 상기 FIFO에 저장된 데이터는 상기 제 2 저장장치로 전달되게되며, 이 과정에서 데이터의 쉬프트가 수행되어진다. 상기 제어 레지스터의 9번 비트의 값이 "0"인 경우 비트 쉬프트를 수행하지 않으며, 상기 제어 레지스터의 9번 비트의 값이 "1"인 경우 상기 비트 쉬프트를 수행한다. 상기 제어 레지스터의 10번 비트는 상기 9번 비트에서 상기 비트 쉬프트를 수행하기로 결정한 경우 상기 쉬프트의 방향을 선택한다. 상기 제어 레지스터의 10번 비트의 값이 "0"인 경우 오른쪽으로 비트 쉬프트를 수행하며, 상기 제어 레지스터의 10번 비트의 값이 "1"인 경우 왼쪽으로 비트 쉬프트를 수행한다.

<29> 상기 제어 레지스터의 11번 비트 내지 13번 비트는 상기 9번 비트에서 쉬프트를 수행하기로 결정한 경우, 쉬프트를 수행할 비트 수를 결정한다. 상기 비트 쉬프트를 수행할 비트 수는 0비트 내지 7비트가 있다. 상기 비트수가 0비트인 경우는 상기 비트 쉬프트를 수행하지 않는 경우와 동일하다. 표 2는 비트 쉬프트 방향이 오른쪽인 경우 상기 소스 어드레스(어드레스 01)의 0번 비트 내지 7번 비트의 데이터를 독출하여 상기 목적지 어드레스(어드레스 41 내지 어드레스 42)에 저장하는 과정을 보이고 있다. 또한 상기 표 2는 상기 전송 사이즈가 8비트인 경우를 가정하며, 상기 어드레스 01에서 독출된 데이터는 어드레스 41에 저장한 이후 어드레스 42에 저장함을 가정한다.

<30>

【표 2】

쉬프트 비트수	독출 데이터의 비트번호(소스 어드레스 01)	저장할 비트번호 (어드레스 41)	독출 데이터의 비트번호(어드레 스 01)	저장할 비트번호(어드레 스 42)
0비트	0번 비트 내지 7 번 비트	0번 비트 내지 7 번 비트		
1비트	0번 비트 내지 6 번 비트	1번 비트 내지 7 번 비트	7번 비트	0번 비트
2비트	0번 비트 내지 5 번 비트	2번 비트 내지 7 번 비트	6번 비트 내지 7 번 비트	0번 비트 내지 1 번 비트
3비트	0번 비트 내지 4 번 비트	3번 비트 내지 7 번 비트	5번 비트 내지 7 번 비트	0번 비트 내지 2 번 비트
4비트	0번 비트 내지 3 번 비트	4번 비트 내지 7 번 비트	4번 비트 내지 7 번 비트	0번 비트 내지 3 번 비트
5비트	0번 비트 내지 2 번 비트	5번 비트 내지 7 번 비트	3번 비트 내지 7 번 비트	0번 비트 내지 4 번 비트
6비트	0번 비트 내지 1 번 비트	6번 비트 내지 7 번 비트	2번 비트 내지 7 번 비트	0번 비트 내지 5 번 비트
7비트	0번 비트	7번 비트	1번 비트 내지 7 번 비트	0번 비트 내지 6 번 비트

<31> 표 3은 비트 쉬프트 방향이 왼쪽인 경우 상기 소스 어드레스(어드레스 01)의 0번 비트 내지 7번 비트의 데이터를 독출하여 상기 목적지 어드레스(어드레스 40 내지 어드레스 41)에 저장하는 과정을 보이고 있다. 또한 상기 표 3은 상기 전송 사이즈가 8비트인 경우를 가정하며, 상기 어드레스 01에서 독출된 데이터는 원래 어드레스 41에 저장함을 가정한다.

<32>

【표 3】

쉬프트 비트수	독출 데이터의 비트번호(소스 어드레스 01)	저장할 비트번호(어드레스 40)	독출 데이터의 비트번호(어드레스 01)	저장할 비트번호(어드레스 41)
0비트			0번 비트 내지 7번 비트	0번 비트 내지 7번 비트
1비트	0번 비트	7번 비트	1번 비트 내지 7번 비트	0번 비트 내지 6번 비트
2비트	0번 비트 내지 1번 비트	6번 비트 내지 7번 비트	2번 비트 내지 7번 비트	0번 비트 내지 5번 비트
3비트	0번 비트 내지 2번 비트	4번 비트 내지 7번 비트	3번 비트 내지 7번 비트	0번 비트 내지 4번 비트
4비트	0번 비트 내지 3번 비트	3번 비트 내지 7번 비트	4번 비트 내지 7번 비트	0번 비트 내지 3번 비트
5비트	0번 비트 내지 4번 비트	2번 비트 내지 7번 비트	5번 비트 내지 7번 비트	0번 비트 내지 2번 비트
6비트	0번 비트 내지 5번 비트	1번 비트 내지 7번 비트	6번 비트 내지 7번 비트	0번 비트 내지 1번 비트
7비트	0번 비트 내지 6번 비트	6번 비트 내지 7번 비트	7번 비트	0번 비트

<33> 표 4는 표 5 내지 표 8에 나타난 데이터와 상기 데이터에 해당하는 값을 2진 값으로 나타낸 나타내고 있다.

<34> 【표 4】

데이터	2진 값	데이터	2진 값
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	F	1110
7	0111	F	1111

<35> 표 5 내지 표 8에서는 소스 어드레스에서 데이터를 독출한 뒤, 상기 비트 쉬프트를 수행하여 상기 목적지 어드레스에 데이터를 기록하는 과정을 보이고 있다. 상기 표 5는 전송비트 단위가 8비트("00")이며, 상기 제어 레지스터의 4번 비트 내지 10번 비트의 값이 "0"인 경우이다. 또한 상기 제어 레지스터의 11번 비트 내지 13번 비트는 4비트 쉬프트 시키는 경우를 나타낸다. 또한 상기 각 어드레스는 8비트의 데이터를 포함하나, 상기 8비트 이외의 사용자가 설정한 임의의 비트 수를 가질 수도 있다.

<36> 【표 5】

소스 어드레스	데이터	목적지 어드레스	데이터
00	1 2	40	U 1
01	3 4	41	2 3
02	5 6	42	4 5
03	7 8	43	6 7
04	9 A	44	8 9
05	B C	45	A B
06	D E	46	C D
07	F 1	47	E F
08	2 3	48	1 2
09	4 5	49	3 4

<37> 상기 표 5에서 보인 바와 같이 상기 소스 어드레스와 목적지 어드레스에 저장되어 있는 데이터들은 8비트이며, 또한 각 어드레스는 2개의 데이터로 이루어져있다. 따라서 상기 표 4에서 설명한 바와 같이 하나의 데이터 값은 4개의 비트로 이루어져있다. 어드레스 00에 저장되어 있는 "1 2"는 상기 목적지 어드레스에 저장될 경우 오른쪽으로 4비트 쉬프트가 수행된 후 저장된다. 따라서 상기 어드레스 00의 데이터 "1"은 상기 목적지 어드레스의 어드레스 40에 기록된다. 이 경우 상기 어드레스 00의 데이터 "1"은 상기 어드레스 40의 후단 4비트에 기록된다. 또한 상기 어드레스 00의 데이터 "2"는 4비트 쉬

프트 됨으로서 상기 어드레스 41에 기록된다. 이 경우 상기 데이터 "2"는 상기 어드레스 41의 전단 4비트에 기록된다. 그리고, 상기 어드레스 40의 전단 4비트는 임의의 값으로 채워진다. 상기 어드레스 01에 저장되어 있는 "3 4"데이터들은 상기 어드레스 41의 후단 부분 내지 어드레스 42의 전단부분에 기록된다. 상기 이와 같은 과정을 수행함으로써 상기 표 5에서 보인 바와 같이 상기 소스 어드레스에 저장되어 있는 데이터 값들이 상기 목적지 어드레스에 기록되게 된다.

<38> 표 6은 전송비트 단위가 32비트("10")이며, 상기 제어 레지스터의 4번 비트 내지 10번 비트의 값이 "0"인 경우이다. 또한 상기 제어 레지스터의 11번 비트 내지 13번 비트는 4비트 쉬프트 시키는 경우를 나타낸다. 또한, 상기 각 어드레스는 8비트의 데이터를 포함하나, 상기 8비트 이외에 사용자가 설정한 임의의 비트 수를 가질 수도 있다.

<39> 【표 6】

소스 어드레스	데이터	목적지 어드레스	데이터
00 내지 03	7 8 5 6 3 4 1 2	40 내지 43	6 7 4 5 2 3 U 1
04 내지 07	F 1 D E B C 9 A	44 내지 47	E F C D A B 8 9
08 내지 09	X X X X 4 5 2 3	48 내지 49	U U U U U U 1 2

<40> 상기 표 6에서 보인 바와 같이 상기 소스 어드레스와 목적지 어드레스에 저장되어 있는 데이터들은 32비트이며, 또한 각 어드레스는 8개의 데이터로 이루어져있다. 따라서 상기 표 4에서 설명한 바와 같이 하나의 데이터 값은 4개의 비트로 이루어져있다. 상기 전송 비트 단위가 32 비트이므로 한번에 독출하는 어드레스는 4개의 어드레스에 저장되어 있는 데이터들이다. 또한 상기 어드레스 00에는 상기 데이터 "1 2"가 저장되어 있으며, 상기 어드레스 01에는 상기 데이터 "3 4"가 저장

되어 있음은 상기 표 5에서 보인 바와 동일하다. 또한 나머지 소스 어드레스에 저장되어 있는 데이터 값들도 상기 표 5와 동일하다. 상기 어드레스 00 내지 어드레스 03에서 독출한 데이터는 상기 어드레스 40 내지 어드레스 44에 4비트 쉬프트되어 기록된다. 상기 어드레스 40 내지 어드레스 49에 저장되는 데이터 값은 상기 표 5와 동일하다.

<41> 표 7은 전송비트 단위가 8비트("00")이며, 상기 제어 레지스터의 4번 비트 내지 9번 비트의 값이 "0"인 경우이다. 또한 상기 10번 비트의 수는 "1"을 나타내며, 상기 제어 레지스터의 11번 비트 내지 13번 비트는 4비트 쉬프트 시키는 경우를 나타낸다. 또한, 상기 각 어드레스는 8비트의 데이터를 포함하나, 상기 8비트 이외에 사용자가 설정한 임의의 비트 수를 가질 수도 있다.

<42> 【표 7】

소스 어드레스	데이터	목적지 어드레스	데이터
00	1 2	40	2 3
01	3 4	41	4 5
02	5 6	42	6 7
03	7 8	43	8 9
04	9 A	44	A B
05	B C	45	C D
06	D E	46	E F
07	F 1	47	1 2
08	2 3	48	3 4
09	4 5	49	5 U

<43> 상기 표 7에서 보인 바와 같이 상기 소스 어드레스와 목적지 어드레스에 저장되어 있는 데이터들은 8비트이며, 또한 각 어드레스는 2개의 데이터로 이루어져있다. 따라서 상기 표 4에서 설명한 바와 같이 하나의 데이터 값은 4개의 비트로 이루어져있다. 어드레스 00에 저장되어 있는 "1 2"는 상기 목적지 어드레스에 저장될 경우 왼쪽으로 4비트 쉬프트가 수행된 후 저장된다. 따라서 상기 어드레스 00의

데이터 "1"은 상기 목적지 어드레스에 기록되지 않는다. 또한 상기 어드레스 00의 데이터 "2"는 어드레스 20의 전단부분에 기록된다. 또한 상기 어드레스 01의 데이터 "3"은 왼쪽으로 쉬프트하여 목적지 어드레스에 저장됨으로서 상기 어드레스 40의 후단부분에 기록에 기록된다. 상기 이와 같은 과정을 수행함으로서 상기 표 7에서 보인 바와 같이 상기 소스 어드레스에 저장되어 있는 데이터 값들이 상기 목적지 어드레스에 기록되게 된다.

<44> 표 8은 전송비트 단위가 32비트("10")이며, 상기 제어 레지스터의 4번 비트 내지 9번 비트의 값이 "0"인 경우이다. 또한 상기 10번 비트의 수는 "1"을 나타내며, 상기 제어 레지스터의 11번 비트 내지 13번 비트는 4비트 쉬프트 시키는 경우를 나타낸다. 또한, 상기 각 어드레스는 8비트의 데이터를 포함하나, 상기 8비트 이외에 사용자가 설정한 임의의 비트 수를 가질 수도 있다.

<45> 【표 8】

소스 어드레스	데이터	목적지 어드레스	데이터
00 내지 03	7 8 5 6 3 4 1 2	40 내지 43	8 9 6 7 4 5 2 3
04 내지 07	F 1 D E B C 9 A	44 내지 47	1 2 E F C D A B
08 내지 09	X X X X 4 5 2 3	48 내지 49	U U U U U 5 3 4

<46> 상기 표 8에서 보인 바와 같이 상기 소스 어드레스와 목적지 어드레스에 저장되어 있는 데이터들은 32비트이며, 또한 각 어드레스는 8개의 데이터로 이루어져있다. 따라서 상기 표 3에서 설명한 바와 같이 하나의 데이터 값은 4개의 비트로 이루어져있다. 상기 전송 비트 단위가 32 비트이므로 한번에 독출하는 어드레스는 4개의 어드레스에 저장되어 있는 데이터들이다. 또한 상기 어드레스 00에는 상기 데이터 "1 2"가 저장되어 있으며, 상기 어드레스 01에는 상기 데이터 "3 4"가 저장

되어 있음은 상기 표 5에서 보인 바와 동일하다. 또한 나머지 소스 어드레스에 저장되어 있는 데이터 값들도 상기 표 5와 동일하다. 상기 어드레스 00 내지 어드레스 03에서 독출한 데이터는 상기 어드레스 40 내지 어드레스 44에 오른쪽으로 4비트 쉬프트되어 기록된다. 상기 어드레스 40 내지 어드레스 49에 저장되는 데이터 값은 상기 표 7과 동일하다.

<47> 도 4는 상기 표 5 내지 표 8에서 설명한 내용을 나타낸 도면이다. 상기 도 5에서 보인 바와 같이 상기 소스 어드레스에 저장되어 있는 데이터들은 4비트씩 쉬프트되어 기록되고 있음을 보이고 있다.

<48> 도 5는 본 발명이 적용되는 상기 제어 레지스터의 비트 값들을 설정하는 과정을 보인 도면이다. 상기 제어 레지스터 설정의 500단계는 상기 제어 명령에 의해 상기 DMA 모드를 결정한다. 상기 DMA 모드는 상술한 바와 같이 상기 하드웨어 모드와 소프트웨어 모드로 이루어져 있다. 상기 제어 레지스터 설정의 502단계는 상기 제어 명령에 의해 상기 전송 사이즈를 결정한다. 상기 제어 레지스터 설정의 504단계는 상기 제어 명령에 의해 소스/목적지 어드레스의 고정 여부를 결정한다. 상기 소스/목적지 어드레스의 고정 여부는 상기 동일한 어드레스에서 반복하여 데이터를 독출하거나, 상기 독출한 데이터를 동일한 어드레스에 반복하여 기록 여부를 결정한다.

<49> 상기 제어 레지스터 설정의 506단계는 상기 604단계에서 다른 소스/목적지 어드레스로부터 데이터를 독출하거나 기록하는 것으로 결정된 경우, 상기 독출하거나 기록할 어드레스를 증가시킬 것인지 감소시킬 것인지 여부를 결정한다. 상기 제

어 레지스터 설정의 508단계는 상기 엔디안 변환 여부를 결정한다. 상기 제어 레지스터 설정의 510단계는 상기 비트 쉬프트 수행 여부를 결정한다. 상기 510단계에서 상기 비트 쉬프트를 수행하기로 결정하면, 512단계 내지 514단계에서 비트 쉬프트의 방향과 비트 쉬프트를 수행 비트 수를 결정한다. 상기 500단계 내지 상기 514단계는 여러 단계로 나누어 수행되어 질 수도 있으나 일반적으로 하나의 과정으로도 수행되어 질 수 있다. 상기 도 5에서 설명한 바와 같이 상기 제어 레지스터의 비트 값들이 설정되고, 상기 제어 레지스터의 0번 비트에 의해 상기 DMA(104)를 구동시키면 데이터의 전송이 이루어진다.

<50> 도 6은 상기 도 5 과정에 의해 상기 제어 레지스터의 비트 값들이 설정되면, 상기 제 1 저장장치(102)에 저장된 데이터를 독출하여 상기 제 2 저장장치(110)로 데이터를 기록하는 과정을 보이고 있다. 물론, 데이터의 전송이 수행되기 이전에 상기 도 2에서 도시되어 있는 상기 SAR, DAR, TCR 등의 레지스터 값들도 설정되어야 한다. 이와 같이 상기 도 2에 도시된 레지스터 값들이 설정되면 데이터 전송이 이루어진다. 상기 도 6의 구성을 보면, 제 1저장장치(102)와 DMA(104), 제 2저장장치(110), 버스 중재기(106), 호스트(114)로 구성되어 있다. 이하, 도 6을 중심으로 상기 제 1저장장치(102)에 저장되어 있는 데이터가 제 2저장장치(110)로 이동되는 과정에 대해 알아본다. 600단계에서 상기 호스트(114)는 제 2저장장치(110)에 저장되어 있는 데이터를 독출한다. 따라서 상기 호스트(114)에 의해 상기 제 2 저장장치(110)의 메모리는 비게 되며, 상기 제 2 저장장치(110)의 비어있는 메모리를 채우기 위해 상기 제 1 저장장치(104)의 데이터를 독출하게 된다. 602단계에서 상기

제 2 저장장치(110)는 상기 DMA(104)로 비어있는 메모리를 채우기 위해 데이터를 요구하게 된다. 상기 제 2 저장장치(110)의 요구에 의해 상기 DMA(104)는 604단계에서 상기 버스 중재기(106)로 상기 제 1 저장장치(102)와의 데이터 송수신에 필요한 버스의 권한을 요구하게 된다. 상기 버스는 한번에 하나의 장치만이 사용할 수 있으며, 동시에 복수 개의 장치로부터 버스 사용의 요구가 수신되는 경우 일정한 순서에 의해 버스 사용을 허가한다. 상기 도 6에서는 상기 DMA(104)가 하나만 도시되어 있지만 실제로 있어서는 복수 개의 DMA에 의해 동작이 이루어진다. 이와 같은 경우 상기 복수 개의 DMA에 의해 복수 개의 버스 사용이 요구될 수 있다. 따라서 상기 버스 중재기(106)는 요구되는 복수 개의 버스 사용을 일정 순서에 따라 허가하게 된다.

<51> 상기 버스 중재기(106)는 606단계에서 상기 DMA(104)의 요구에 의해 사용할 버스를 제공한다. 상기 제공된 버스에 의해 상기 DMA(104)는 608단계에서 상기 제 1 저장장치(102)에 저장되어 있는 데이터를 독출한다. 상기 독출한 데이터는 상기 DMA(104) 내부에 존재하고 임시 메모리(FIFO)에 저장된다. 상기 제 1 저장장치(102)로부터 독출할 데이터의 어드레스는 상기 도 2에서 설명한 바와 같이 상기 DMA 내부에 위치하고 있는 상기 SAR에 의해 이루어진다. 상기 제 1 저장장치(102)에 저장되어 있는 데이터를 독출하여 상기 FIFO에 저장하면 상기 DMA는 610단계에서 상기 제 2 저장장치(110)와 데이터 송수신에 필요한 버스를 상기 버스 중재기(106)로 요구한다. 이하 상기 610단계 내지 612단계에서 이루어지는 과정은 상기 604단계 내지 606단계에서 이루어지는 과정과 동일하다. 상기 버스 중재기(106)로부터 상기 버스 사용을 허락받으면 상기 DMA(104)는 상기 FIFO에 저장되어 있는 데이터를 상기 제 2 저장장치(110)로 전달한다. 상기 독출한 데이터를 저장할 어드레스는 상기 도 2에서 설명한 바와 같이 상기 DAR에 의해 이루어진다.

<52> 상기 DMA(104)는 상기 600단계 내지 614단계를 반복 수행함으로써 상기 제 1 저장장치(102)에 저장되어 있는 데이터를 상기 제 2 저장장치(110)에 전달하게 된다. 또한 상기 반복 수행의 회수는 상기 도2에서 설명한 바와 같이 TCR에 의해 이루어진다. 상기 TCR에 설정된 회수에 의해 데이터 전송이 수행되면 상기 CPU 또는 외부 제어 장치로 상기 데이터 전송이 끝났음을 알리게 된다.

<53> 도 7은 무선 접속 제어(Radio Link Control: RLC) 계층이나 매체 접근 제어(Medium Access Control: MAC)계층에서의 데이터 전송 과정을 보이고 있다. 상기 RLC 계층이나 MAC 계층에서 헤더를 붙이고자 할 경우 상기 헤더만큼 비트 수가 쉬프트되어 전송됨이 요구된다. 또한 상기 비트 쉬프트가 상기 CPU에 의해 이루어지는 것보다 본 발명에서 제안하는 DMA에 의해 수행되는 것이 처리속도에 있어 효과적이다. 표 9는 상기 CPU에 의한 데이터 처리속도와 상기 DMA에 의한 데이터 처리속도를 비교하고 있다.

<54> 【표 9】

데이터 크기	캐쉬	CPU에 의한 처리	CPU+DMA에 의한 처리	DMA에 의한 처리
1k × 6비트	구동	872.2	1028	242.4
	구동하지 않음	8576.2	8836	269.2

【발명의 효과】

<55> 전술한 바와 같이 본 발명은 종래 CPU에 의해 특정 비트 수만큼 쉬프트된 데이터를 전송함으로써 발생하는 시간의 지연문제를 DMA에 의해 특정 비트 수만큼 쉬프트된 데이

터를 전송함으로서 시간의 지연문제를 해결하였다. 또한 CPu에 의해 데이터를 처리함으로서 발생하는 CPU의 과부하를 DMA에서 처리함으로서 CPU의 과부하 문제를 해결하였다.

【특허청구범위】**【청구항 1】**

직접 메모리 접근매체에 의해 데이터를 독출 및 저장하는 방법에 있어서,

제 1저장매체로부터 독출한 데이터의 가공이 요구될 시 쉬프트 방향과 쉬프트 하고자 하는 비트 수를 미리 결정하는 과정과,

상기 독출한 데이터를 구성하는 비트들의 열을 레지스터에 순차적으로 저장하고, 상기 저장한 비트들의 열을 상기 결정한 쉬프트 방향으로 상기 쉬프트 하고자 하는 비트 수만큼을 쉬프트 하여 제2저장매체로 전달하는 과정을 포함함을 특징으로 하는 상기 방법.

【청구항 2】

제1항에 있어서, 상기 독출한 데이터를 구성하는 비트들의 열은 8비트, 16비트, 32비트 중 어느 하나로 이루어짐을 특징으로 하는 상기 방법.

【청구항 3】

제 2항에 있어서, 상기 쉬프트 할 비트 수는 0 비트 내지 7 비트 중 하나의 비트 수로 쉬프트함을 특징으로 하는 상기 방법.

【청구항 4】

제 3항에 있어서, 상기 쉬프트 방향과 쉬프트 하고자 하는 비트 수를 결정하는 과정은, 상기 직접 메모리 접근매체에 설정되어 있는 비트 값들에 의해 결정됨을 특징으로 하는 상기 방법.

【청구항 5】

제1항에 있어서, 상기 독출한 데이터를 구성하는 비트들의 열을 중요도가 높은 비트들과 중요도가 낮은 비트들의 열로 구분하는 과정과,

상기 중요도가 낮은 비트들과 상기 중요도가 높은 비트들의 위치를 재배열하여 상기 제 2저장매체에 기록하는 과정을 부가함을 특징으로 하는 상기 방법.

【청구항 6】

제 5항에 있어서, 상기 중요도가 높은 비트들과 상기 중요도가 낮은 비트들의 위치를 재배열하는 과정은 상기 독출한 데이터가 32비트로 구성된 비트들임을 특징으로 하는 상기 방법.

【청구항 7】

제 6항에 있어서, 상기 중요도가 높은 비트들과 상기 중요도가 낮은 비트들의 위치를 재배열을 결정하는 과정은, 상기 직접 메모리 접근매체에 설정되어 있는 비트 값에 의해 결정됨을 특징으로 하는 상기 방법.

【청구항 8】

직접 메모리 접근매체에 의해 데이터를 독출 및 저장하는 장치에 있어서,
소스 어드레스에 지시한 독출 데이터를 저장하고 있는 제 1저장매체와,
독출 데이터의 가공이 요구될 시 쉬프트 방향과 쉬프트 하고자 하는 비트 수를 미리 결정하고, 상기 독출한 데이터를 구성하는 비트들의 열을 레지스터에 순차적으로 저장하고, 상기 저장한 비트들의 열을 상기 결정한 쉬프트 방향으로 상기 쉬프트 하고자 하는 비트 수만큼을 쉬프트 하여 전달하는 상기 직접 메모리 접근매체와,
상기 직접 메모리 접근매체로부터 전달된 데이터를 저장하는 제 2저장매체로 구성됨을 특징으로 하는 상기 장치.

【청구항 9】

제8항에 있어서, 상기 직접 메모리 접근매체는,
독출한 데이터를 구성하는 비트들의 열을 8비트, 16비트, 32비트 중 어느 하나의 비트로 독출함을 특징으로 하는 상기 장치.

【청구항 10】

제 9항에 있어서, 상기 직접 메모리 접근매체는,
상기 쉬프트 할 비트 수는 0 비트 내지 7 비트 중 하나의 비트 수로 쉬프트함을 특징으로 하는 상기 장치.

【청구항 11】

제 10항에 있어서, 상기 직접 메모리 접근매체는,
설정되어 있는 비트 값들에 의해 상기 쉬프트 방향과 쉬프트 하고자 하는 비트 수를 결정함을 특징으로 하는 상기 장치.

【청구항 12】

제 8항에 있어서, 상기 직접 메모리 접근매체는,
독출한 데이터를 구성하는 비트들의 열을 중요도가 높은 비트들과 중요도가 낮은 비트들의 열로 구분하고, 상기 중요도가 낮은 비트들과 상기 중요도가 높은 비트들의 위치를 재배열하여 상기 제 2저장매체에 기록함을 특징으로 하는 상기 장치.

【청구항 13】

제 12항에 있어서, 상기 직접 메모리 접근매체는,

상기 독출한 데이터가 32비트로 구성된 경우 상기 중요도가 높은 비트들과 상기 중요도가 낮은 비트들의 위치를 재배열함을 특징으로 하는 상기 장치.

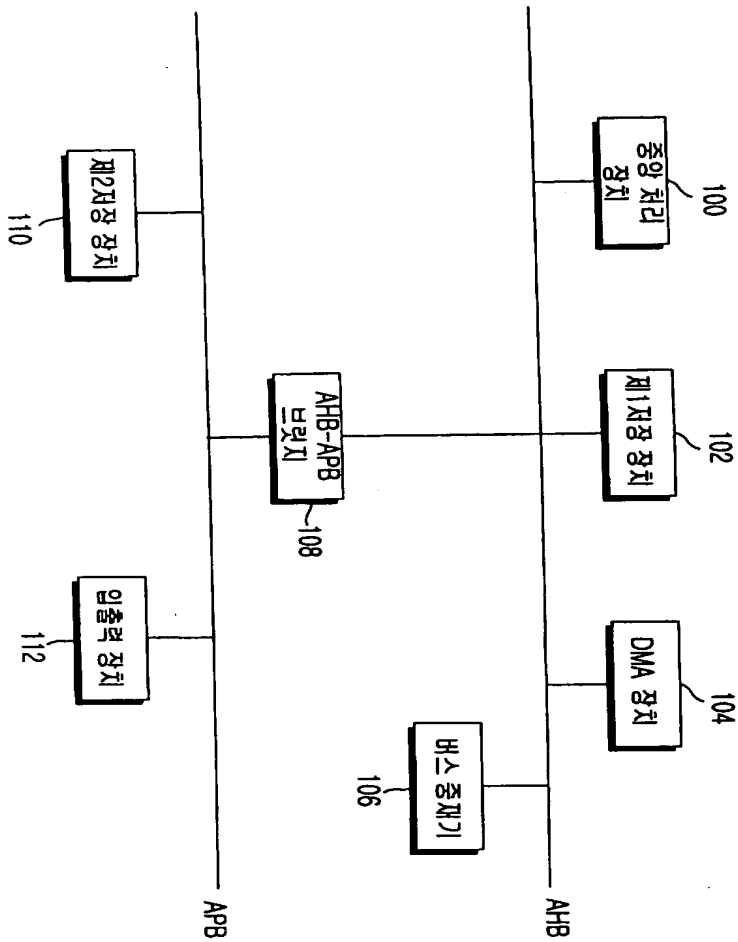
【청구항 14】

제 13항에 있어서, 상기 직접 메모리 접근매체는,

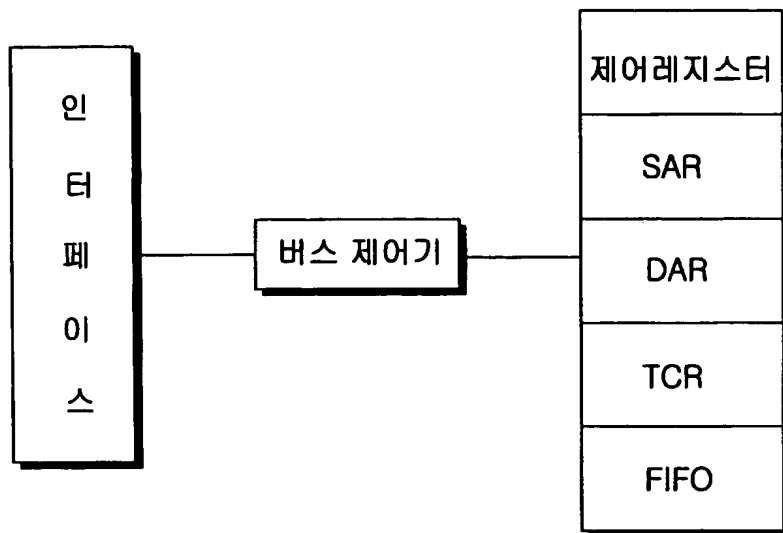
설정되어 있는 비트 값들에 의해 상기 중요도가 높은 비트들과 상기 중요도가 낮은 비트들의 위치를 재배열함을 결정하는 상기 장치.

【도면】

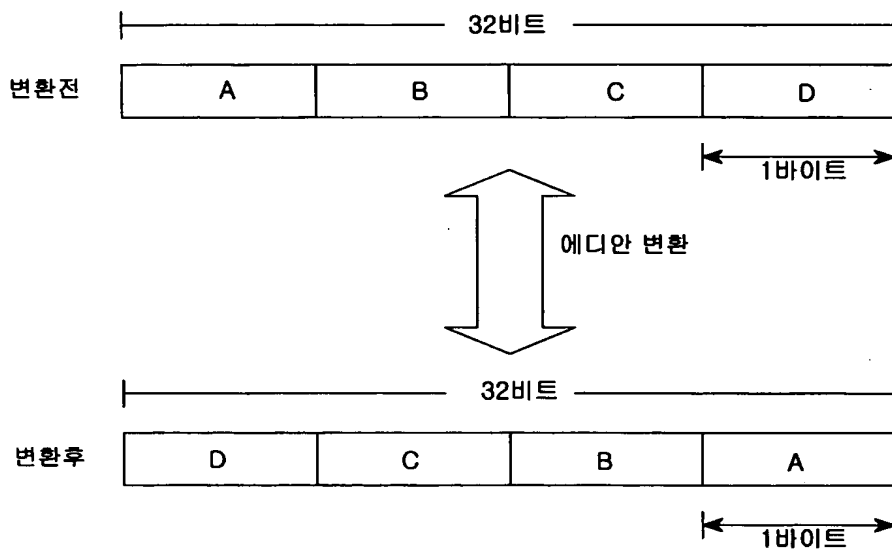
【도 1】



【도 2】



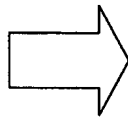
【도 3】



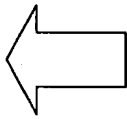
【부 4】

데이터
목적지 주소

U	12	34	56	78	9A	BC	DE	F1	23	4
40	41	42	43	44	45	46	47	48	49	

오른쪽으로 4비트
쉬프트데이터
소스 주소

1	23	45	67	89	A8	CD	EF	12	34	5
00	01	02	03	04	05	06	07	08	09	

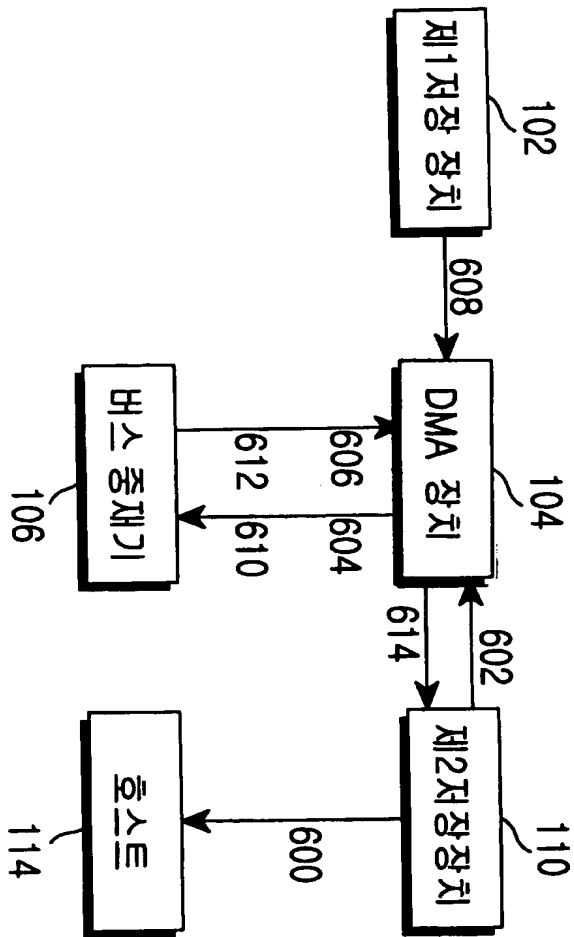
왼쪽으로 4비트
쉬프트데이터
목적지 주소

2	34	56	78	9A	BC	DE	F1	23	45	U
40	41	42	43	44	45	46	47	48	49	

【도 5】



【도 6】



【도 7】

